組込みシステムのセキュリティへの取組みガイド

15個の具体的なチェック項目により,
自組織のセキュリティレベルを明確にする

2009年6月

独立行政法人 情報処理推進機構
セキュリティセンター

目 次

図 目 次

表 目 次

ガイド公開に当たって

近年、多数多種の組込み機器をつなぐことのできるネットワーク環境が整いつつある。今後も組込みシステムは更なる高機能化・多機能化の方向で進化し、社会のあらゆるインフラの中に組み込まれていくことが予想される。

一方、パソコンやサーバで利用されている汎用的なソフトウェアやハードウェアが、コスト低減、開発期間短縮、生産性向上の面では組込みシステムの開発に対して有利に適用されるが、品質面や安全性での要求レベルにうまく対応できるかが大きな懸念となっている。さらに、ネットワーク化の結果として、現在のパソコンが直面しているセキュリティの脅威に、組込みシステムが直面していくことになる。悪意のある攻撃者にとって金銭情報に密接に関与している組込みシステムは攻撃の対象になっている。また、社会の混乱を起こそうとする悪意のある攻撃者にとっても、組込みシステムは社会インフラを支える重要な構成要素となっているので攻撃の対象になるであろう。

組込みシステムの開発者が、パソコンでのセキュリティ・インシデントの経験を踏まえた組込みシステム向けセキュリティ対策を実施できるような、アプローチの確立とその普及において、啓発がますます重要になっている。

セキュリティへの対策には100%完璧であるということはなく、投資とのバランスで決められることが多い。このバランスとセキュリティへの投資をきちんと把握するためには、自組織の現状の「セキュリティへの取組み」のレベルを正しく認識することが必要になる。特に組込みシステムのプロジェクトでは、開発スケジュールなどが極めてタイトであったり、開発途中で要求仕様や制約条件が変わったりする場合が多く発生する。これらを解決するためにセキュリティ上の課題を整理し、製品のセキュリティを向上していくことを目的とした具体的な「セキュリティへの取組み」をまとめたガイドが必要である。

組込みシステムの開発者はこのガイドを参照することで、自組織がどのレベルに位置しているかを把握でき、さらに上位のレベルに向けた取組みの指針を得ることができる。

今回、IPAでは科学技術振興調整費プロジェクト「組込みシステム向け情報セキュリティ技術」で、上記目的のガイドの整備をすることにした。

事故を引き起こさない信頼性（Reliability）と安全性（Safety）の追求に加え、ネットワークや人的経由での悪意の攻撃にも対処できるセキュリティ（Security）対策の整備・強化が、両輪として議論され、対策されていくことが急務である。

独立行政法人情報処理推進機構（IPA）セキュリティセンター
情報セキュリティ技術ラボラトリー長　小林　偉昭

独立行政法人情報処理推進機構（株式会社ユビテック※）　赤羽　千英
独立行政法人情報処理推進機構（株式会社日立製作所※）　受田　賢知
独立行政法人情報処理推進機構（株式会社日立製作所※）　萱島　信
独立行政法人情報処理推進機構（株式会社IT働楽研究所※）　添野　元秀
独立行政法人情報処理推進機構（凸版印刷株式会社※）　高橋　聡
独立行政法人情報処理推進機構　中野　学

※2009年6月現在

1. はじめに

1.1. 組込みセキュリティの現状と課題

マイクロプロセッサやメモリ等の半導体性能の向上に伴い、産業機器や家電製品は益々高機能化の一途を辿っている。それらの機器を制御するために機器の内部に組み込まれたコンピュータシステム、「組込みシステム」が様々な分野で利用されている。また、情報通信技術の進展により、これらの組込みシステムがインターネットなどのオープンなネットワークに接続されるようになりつつあり、パーソナルコンピュータと同様に、不正な利用を試みる第三者の攻撃ターゲットになる可能性が高くなってきた。

ハイビジョン放送やWebショッピング、高機能な体重計を利用したプライバシー情報の管理など、様々なサービスの拡大に伴い、組込みシステムで取り扱う情報の価値も向上している。そのため、組込みシステムの開発においては、外部からの攻撃や不正なデータ複製への対策、廃棄時のプライバシー情報の削除等を考慮した実装が求められるようになってきている。

しかし、組込みシステムの開発現場では、市場における価格競争の激化を要因とする開発期間の極端な短縮化のために、セキュリティへの対応が疎かにされやすい現状がある。開発の最前線にいるエンジニアをはじめとして、責任者や経営陣がセキュリティに対してどのように取り組んでいくべきかが課題となっている。

1.2. 本書の狙い

そこで本書では、組込みシステムの開発元を対象として、よりセキュアな組込みシステムを開発するために、(１）開発チームのマネジメント面、（２）組込みシステムの企画・開発面、（３）組込みシステムの利用時において開発元が行うべき運用面、（４）組込みシステムの廃棄面に関して行うべき取組み事項のガイドを示すこととした。本書は組込みシステムを開発する企業に所属する、以下の者を対象として想定している。

- 開発プロジェクトの開発者
- 開発プロジェクトを担当する開発責任者
- 開発プロジェクトの予算や人員を決定する意志決定者（経営層）

本書の狙いは以下の通りである。

（１） 組込みシステム開発関係者のセキュリティ意識向上

組込みシステムに求められるセキュリティとは何かを理解した上で、現在の開発におけるセキュリティへの取組み内容を認識し、開発現場で具体的に実践できる対策手法を身につけられる。

（２） よりセキュアな組込みシステムの実装

情報化社会に向けて、ユーザ自身への被害の防止はもちろんのこと、ユーザが使用中の組込み機器が踏み台として悪用されることにより発生する、第三者に対する被害の防止にも役立つ。このことは、セキュリティ事故の発生による費用負担や、企業ブランド価値の低下の防止にもつながる。

2.　用語の定義

・　組込みシステム

産業機器や家電製品などの機器に組み込まれた、半導体や周辺装置からなる部分で、機器の制御を行う目的に専用化されたコンピュータシステムのこと。

・　組込み機器

組込みシステムによって制御される産業機器や家電製品などのこと。

・　脆弱（ぜいじゃく）性

組込みシステムにおいて、不正アクセスやコンピュータウィルス等の攻撃により、その機能や性能を損なう原因となり得るセキュリティ上の問題箇所をさす。

・　セキュリティルール

組織におけるセキュリティに対する取組みが、場当たり的にならないようにすることを目的として、具体的に行うべきこと、および行ってはならないことを定めたセキュリティに対する取組み方針をさす。

3. セキュリティへの取組みのレベルとは

3.1. 組込みシステムとセキュリティ

3.1.1. 組込みシステムの内部構成

組込みシステムは、制御対象である機器に応じて、ワンチップマイコンで実現した小規模なものからプロセッサを複数個用いて構成される大規模なものまで様々である。このため本書では、図 3-1のようにシステムの構成モジュールや外部とのインタフェースをモデル化した上で、組込みシステムのセキュリティ対策手法を検討することにした。
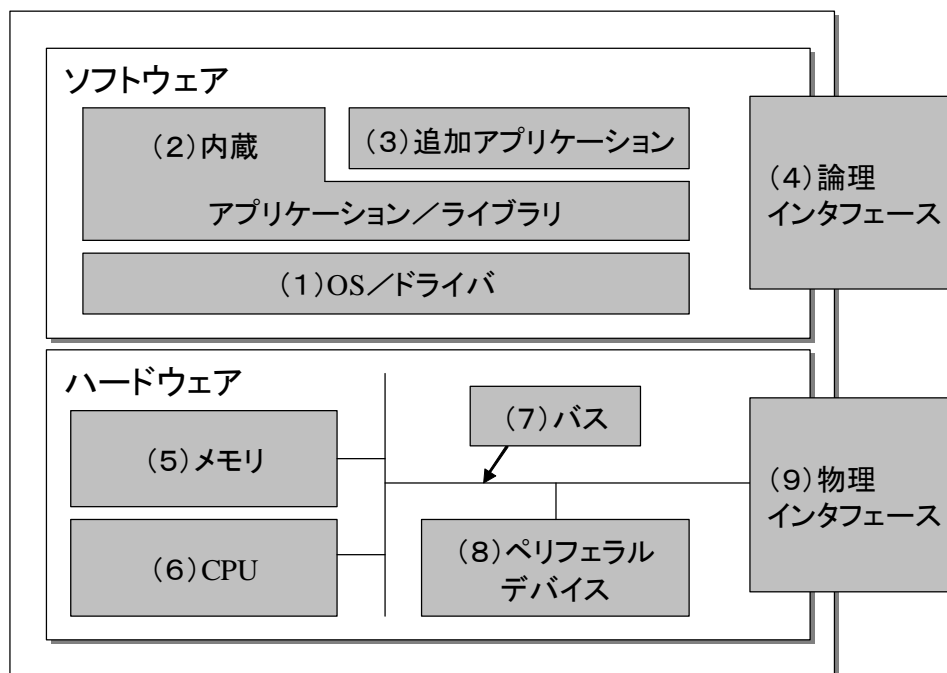
図 **3-1** 組込みシステムの内部構成モデル

組込みシステムは、何らかのOS／ドライバ（図中の（１））と、その上で動作する内蔵アプリケーション／ライブラリ（図中の（２））および、ユーザが任意に付加できる追加アプリケーション（図中の（３））で構成されるソフトウェアと、ソフトウェアの動作基盤となるハードウェアで構成されている。ハードウェアは、メモリ（図中の（５））とCPU（図中の（６））、TPM（Trusted Platform Module）等のペリフェラルデバイス（図中の（８））および、それらを結合するバス（図中の（７））により構成されている。

組込みシステムは、各種コネクタ、ネットワーク端子、タッチパネルディスプレイ等の物理インタフェース（図中の（９））と、各種プロトコルの論理インタフェース（図中の（４））を組み合わせてユーザや他システムと接続されている。

3.1.2.　組込みシステムに対する攻撃と対策技術

近年組込みシステムは、インターネットへの接続を前提とした使われ方をするものが多くなっており、パーソナルコンピュータなどと同様に攻撃のターゲットになる可能性が高まっている。ユーザが組込みシステムを安全に使用するには、少なくとも以下の3つの指針に基づいた対策を行うことが求められる。

機密性（Confidentiality）	承認されていない人に、システム内部の機密情報が開示されないようにすること
一貫性（完全性）（Integrity）	保管中、処理中、または転送中に、情報が改ざんされていないようにすること
可用性（Availability）	システムが適切に稼働し、許可されたユーザがサービスを利用できることを保証すること

組込みシステムに対する脅威は、論理および物理インタフェースを介して行われる攻撃と、組込みシステムを入手した攻撃者がソフトウェアおよびハードウェアを解析することにより行われる攻撃が考えられる。これらの性質を実現するには、3.1.1節で説明した組込みシステムを構成する各モジュールそれぞれに対し、発生する可能性のある脅威をSTRIDE手法[1] 等を用いて分析すると共に、対策を検討する必要がある。表 3-1に組込みシステムを構成するモジュールごとに発生しうる脅威と対策技術を整理した結果を示す。

表 3-1 組込みシステムで発生しうる脅威と対策技術

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">機密性</th><th colspan="2">一貫性（完全性）</th><th colspan="2">可用性</th></tr>
<tr><th>脅威</th><th>対策技術</th><th>脅威</th><th>対策技術</th><th>脅威</th><th>対策技術</th></tr>
<tr><td rowspan="4">ソフトウェア</td><td>論理インタフェース</td><td>通信時のなりすまし</td><td>ユーザ認証</td><td>通信時のデータ改ざん</td><td>データ署名</td><td>サービス拒否攻撃</td><td>フィルタリング</td></tr>
<tr><td>追加アプリケーション</td><td>マルウェアのインストール</td><td>証明書検証</td><td rowspan="2">ソフトウェアの改ざん</td><td rowspan="2">脆弱性対策
ロギング</td><td rowspan="2">サービス拒否攻撃</td><td rowspan="2">フィルタリング
脆弱性対策
ロギング</td></tr>
<tr><td>内蔵アプリケーション／ライブラリ</td><td>特権昇格
ソフトウェア構成の解読</td><td>脆弱性対策
ソフトウェア難読化</td></tr>
<tr><td>OS／ドライバ</td><td>特権昇格</td><td>脆弱性対策</td><td>特権昇格</td><td>脆弱性対策
ロギング</td><td>サービス拒否攻撃</td><td>脆弱性対策
ロギング</td></tr>
<tr><td rowspan="5">ハードウェア</td><td>メモリ</td><td rowspan="5">プロービング</td><td rowspan="5">暗号化
耐タンパ機能の実装</td><td rowspan="5">データ改ざん
データ破壊</td><td rowspan="5">耐タンパ機能の実装</td><td rowspan="5" colspan="2">（一貫性が損なわれることにより発生する）</td></tr>
<tr><td>CPU</td></tr>
<tr><td>バス</td></tr>
<tr><td>ペリフェラルデバイス</td></tr>
<tr><td>物理インタフェース</td></tr>
</table>

これらの対策は、出荷前にその機能が実装されているだけでなく、新たに発覚した脅威にも対応できるように、出荷後にもその機能が維持されることが重要である。

[1] （米）Microsoft社のSecurity Engineering and Communicationsグループが策定した、脅威の種別をSpoofing (なりすまし)、Tampering (改ざん)、Repudiation (否認)、Information Disclosure (情報漏えい)、Denial of Service (サービス拒否)、および Elevation of Privilege (特権の昇格)に分類する分析手法

3.2. 組込みシステムのライフサイクル

3.2.1. ライフサイクルの定義

3.1節で述べたように、組込みシステムのセキュリティを向上させるためには、取り扱う情報資産の価値に応じた適切なセキュリティ対策を、情報を処理または伝達する各部位に対して実施することが求められる。その際には組込みシステムのライフサイクルを通した分析が有効となる[2) 3]。本書では、このライフサイクルを、「企画」、「開発」、「運用」、「廃棄」の４つのフェーズに分け、ライフサイクル全体に必要な、製造メーカとしてマネジメントすべき事項について説明を行っていく。

例えば情報家電であれば、製造メーカにより企画・開発され、小売業者を経てユーザの手に渡り実際に運用されたのち、リサイクル業者によって廃棄される。運用においてユーザの手に渡ると、まず初期設定されてから実際に利用が開始される。利用中にはソフトウェアの更新が行われることや、譲渡・売却等によって他のユーザの手に渡り、あらためて初期設定されることもある。組込みシステムに対する脅威は、運用フェーズだけでなく、その他のフェーズにおいても存在するため、それらのセキュリティ課題はあわせて解決しなければならない（図 3-2）。
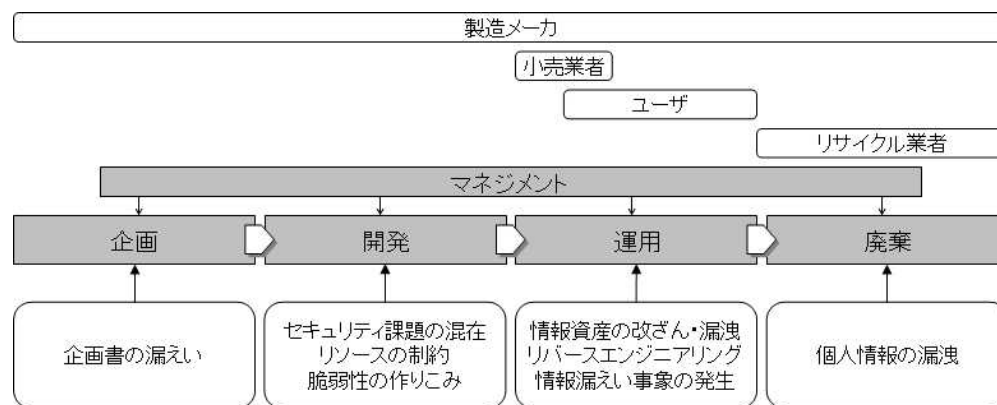
図 3-2 ライフサイクルの各フェーズにおけるセキュリティ課題の例

以下ではまず、ライフサイクルの各フェーズの役割と、それぞれのフェーズにおけるセキュリティの考慮点を大まかにまとめる。

（1） 企画フェーズ

組込みシステムの企画を行い、開発における予算について検討するフェーズである。この段階でどの程度、製品のセキュリティを重視するかの「セキュリティへの取組み」のレベルが決定される。企画フェーズではライフサイクル全体のセキュリティに関する準備が必要となる。そのため、製品のセキュリティの重視度合いによって、開発や運用、廃棄時のためにどの程度の予算を配置し、振り分けるかを、あわせて検討する必要がある。

[2)] 組込みシステムの脅威と対策に関するセキュリティ技術マップの調査報告書
（http://www.ipa.go.jp/security/fy18/reports/embedded/index.html）

[3] 複数の組込み機器の組み合わせに関するセキュリティ調査報告書
（http://www.ipa.go.jp/security/fy19/reports/embedded/index.html）

（２） 開発フェーズ

製品の利用目的である要求仕様に基づいて、ハードウェアやソフトウェアの設計が行われる。特にソフトウェアの設計および実装の段階において脆弱性が作り込まれやすく、本書では具体的な対策手法を挙げる。また開発の外部委託や品質評価の過程、工場での組立て作業に至るまで、製造元が製品を出荷するために考慮すべきセキュリティへの対策内容についても触れる。開発フェーズにおけるセキュリティ対策の目的は、情報資産の保護であると言えるだろう。表 3-2に情報資産の例を示す。

表 3-2 組込みシステムで保護すべき情報資産の例

情報資産	説明
コンテンツ	音声、画像、動画等のマルチメディアデータ（商用コンテンツ利用時の著作権管理データおよびプライベートコンテンツ等）、コンテンツ利用履歴（コンテンツの利用履歴も保護することが重要）等
ユーザ情報	ユーザの個人情報（氏名/住所/電話番号/生年月日/クレジットカード番号等）、ユーザ認証情報、利用履歴等
機器情報	情報家電そのものに関する情報（機種、ID（Identification）、シリアルID等）、機器認証情報等
ソフトウェアの状態データ	各ソフトウェアに固有の状態データ（動作状態、ネットワーク利用状態等）
ソフトウェアの設定データ	各ソフトウェアに固有の設定データ（動作設定、ネットワーク設定、権限設定、バージョン等）
ソフトウェア	OS（Operating System）、ミドルウェア、アプリケーション等（ファームウェアと呼ばれることもある）
設計データ内部ロジック	企画・設計フェーズで発生する仕様書・設計書等の設計情報

（３） 運用フェーズ

組込みシステムは、ユーザが家電量販店などを通じて入手して利用する。ユーザは自身や自宅の個人情報を情報家電内に記録し、さらに音声、音楽、画像、映像等のプライベートコンテンツを機器内に保存して利用するようになり、これらの個人情報やコンテンツデータがさらに保護すべき情報資産として増えていくことになる。

組込みシステムは、その運用フェーズにおいて、（１）ユーザの宅内に設置する（第三者が直接、触れる機会が少ない）、（２）家庭内ネットワークと接続して通信する、（３）携帯電話等の携帯端末と連携する場合がある、といった特徴があり、セキュリティ上これらの点も各フェーズで考慮する必要がある。

また組込みシステムの出荷後でも脆弱性が発見されることがあるため、対策として組込みシステム内部のプログラム更新などの機能を必要とする。

（４） 廃棄フェーズ

組込みシステムは買い替えや利用停止、故障を機会として廃棄されることがある。廃棄の際に、個人情報などの機密データを消去するための方法を、利用者に周知させる仕組みが必要である。

3.3. セキュリティへの取組みのレベルとその概要

前節までにおいて、組込みシステムのセキュリティをシステムの内部構成とライフサイクルという側面から捉え、脅威と対策について述べてきた。しかしながら、このように脅威と対策を列記しても、「何をすべきか」を具体化しにくいのが実状である。その原因の一つに現場と経営層の意識の違いがある。例えば経営層はセキュリティを現場の問題だと考えて具体的な方針を定めず、開発責任者や開発者は十分な投資や開発期間がないためセキュリティを棚上げにしている場合がある。しかし現在の大規模化した組込みシステムにおいて、セキュリティリスクのない製品を想定することは難しく、上流の工程で目を反らしていた問題は、より深刻な形で下流の工程に影響を及ぼすことになる。こういった問題は、経営層・現場の双方が組込みシステムに関わる脅威を正しく認識し、共に努力することでしか解決には至れない。本書では、各企業の現状を可視化するために、まずセキュリティに対する取組みを方針ごとに４フェーズに分類した（表 3-3）。開発責任者や開発者が、自社の取組み方針に近い水準の施策を参考にすることで、現状の再認識や課題抽出の一助となることを期待している。以下では、組込みシステムのライフサイクルの各フェーズと、それらのフェーズに必要なマネジメントにおける取り組み方針についての説明を行っていく。また各方針に基づいたフェーズごとの詳細な取組み内容については4章に記載した。

表 3-3 ライフサイクルにおける「セキュリティへの取組み」のレベル

	マネジメント方針	企画方針	開発方針	運用方針	廃棄方針
レベル4	セキュリティへの取組みに対して、組織としての方針策定および実施に加え、監査のプロセスを有する	組織の方針に基づき、客観的評価を想定したセキュリティルールが策定および運用される		組織の方針として脆弱性対策方針を定めると共に、一般への公開を行う	客観的な基準に基づいたセキュリティリスクの軽減方法が用意される
レベル3	セキュリティへの取組みに対して、組織としての方針策定および実施する	組織の方針に基づいてセキュリティルールが策定および運用される		組織の方針として、脆弱性対策方針を定めている	廃棄時のセキュリティリスクの軽減方法が用意される
レベル2	セキュリティへの取組みを開発責任者や開発者に一任し、問題を案件ごとに個別に対応する	セキュリティルールが開発責任者や開発者主導で策定および運用される		脆弱性対策方針を製品ごとに定めている	廃棄方法が仕様書等に明記される
レベル1	セキュリティへの取組みを行っていない	セキュリティルールは定めていない		脆弱性に対する保証基準を設けていない	廃棄方法が考慮されていない

3.3.1. マネジメント方針

マネジメント方針は、経営層のセキュリティに対する取組み方針を指す。ここでは、経営層と、開発現場の間でどのように意識の共有や方針の策定および実施が行なわれているかが主な基準となる。マネジメントに関する各レベルの取組み内容をまとめると以下のようになる。

- レベル１

  セキュリティへの取組みを行なっていない水準である。経営層と開発現場の認識に差があり、経営層はセキュリティを現場の問題だと考えて具体的な方針を定めず、開発現場は十分な投資や開発期

間がないためセキュリティを棚上げにしている場合等が該当する。そのため製品の品質は、セキュリティ上の問題が顕現しやすいタイプの製品であるかどうかに強く依存することになる。

・ レベル２

セキュリティへの取組みを開発現場に一任し、問題を案件ごとに個別に処理する水準である。経営層と開発現場が共にセキュリティを組織として取り組む問題だと認識しているものの、具体的な方策がなく、実質的に開発現場任せになっている場合等が該当する。そのため開発現場によって品質に差が生じ、またある製品で培った経験が他の製品で活かされにくいという問題がある。

・ レベル３

セキュリティへの取組みに対して組織としての方針を策定および実施している水準である。経営層と開発現場は問題意識を共有しており、方針をドキュメントとしてまとめたり、開発現場へのセキュリティ教育を実施し、現場で起こった問題の知識共有等を行なったりしている場合が該当する。組織としての基準があることから、製品品質の底上げが期待できるが、方針自体の過不足や、基準を正しく満たしているかどうかの確認など、客観的な視点に立った評価という側面ではまだ課題がある。

・ レベル４

セキュリティへの取組みに対して組織としての方針策定および実施に加え、監査のプロセスを有する水準である。経営層と開発現場は、製品を開発する上での社会的責任や役割を認識し、客観的に自社の取組みを評価する仕組みを導入したり、そのための組織やフロア・工場を整備したりする場合等が該当する。この水準に達するためには経営層と開発現場が一丸となってセキュリティに取り組むことが求められ、本書における一つの目標となる水準である。

3.3.2. 企画・開発方針

「企画・開発方針」は、製品を企画・開発する上でのセキュリティの取組み方針である。ここでは、セキュリティルールの策定と、セキュリティルールに基づいた開発が十分に行なわれているかが主な基準となる。企画、開発に関する各レベルの取組み内容をまとめると以下のようになる。

・ レベル１

セキュリティルールが定められていない水準である。セキュリティ製品を開発する上での基準を設けておらず、企画フェーズで特にセキュリティリスクの検討を行なっていない場合などが該当する。このことから、品質は設計仕様書に強く依存し、評価や運用のフェーズになって初めてセキュリティ上の問題が顕現化することが多い。またあらかじめ想定していない種類の問題であるため、対応に非常に時間がかかる。

・ レベル２

セキュリティルールが担当者主導で策定および運用されている水準である。製品の脆弱性等の問題

を解決するためにはセキュリティ実装や専用の評価手法が必要になることを担当者が認識しており、企画フェーズでどういう問題が発生するかを十分に検討している場合等が該当する。そのため、担当者の技術や知識に依存するものの、仕様書作成のフェーズで目立ったセキュリティリスクは排除可能となる。

・ レベル３

組織の方針に基づいてセキュリティルールが策定および運用される水準である。脆弱性分析やセキュリティ実装、セキュリティ評価にはある程度形式的なやり方があるものの、網羅的に実施することは現実的ではないという理由から、組織の方針に基づき各項目の優先度を担当者が把握した上でセキュリティルールを決めている場合等が該当する。このレベルでは、担当者へのセキュリティ教育は不可欠となるが、大部分のセキュリティリスクを上流で取り除くことが期待できる。

・ レベル４

組織の方針に基づき、客観的評価を想定したセキュリティルールが策定および運用される水準である。担当者はセキュリティ教育を通じて、ITセキュリティ評価および認証制度（JISEC：Japan Information Technology Security Evaluation and Certification Scheme）や暗号モジュール試験および認証制度(JCMVP：Japan Cryptographic Module Validation Program)等の認証制度で利用されている様式についての知見を得て、社外に通用するセキュリティルールを策定すると共に、専門の部署と連携して脆弱性関連情報の収集や製品のセキュリティルールの適合度を評価する場合等が該当する。この水準では、問題が起きた場合に形式的に原因の調査と報告が行えるため、迅速に問題の解決が可能となる、本書における一つの目標となる水準である。

3.3.3. 運用方針

「運用方針」は、製品出荷後に発見された脆弱性に対する取組み方針である。ここでは、脆弱性対策方針が主な基準となる。運用に関する各レベルの取組み内容をまとめると以下のようになる。

・ レベル１

脆弱性に対する保証基準を設けていない水準である。内部での情報共有や自発的な顧客への通知が行なわれることはなく、クレームがあった場合に程度に応じて仕様の一部と判断するか、対策を行なうかを判断して通知する場合等が該当する。

・ レベル２

脆弱性対策方針を開発責任者や開発者が製品ごとに決めている水準である。開発責任者や開発者は製品の脆弱性を整理し、深刻な場合にはユーザに対して公開している場合などが該当する。このため、製品の脆弱性を意識しないユーザはソフトウェアの更新があっても気づかずに製品を使い続け、問題が拡大する場合もある。また、脆弱性関連情報が組織内でうまく共有されていないため、同じ問題が繰り返される場合もある。

- レベル３

組織の方針として、脆弱性対策方針が含まれている水準である。脆弱性は全社的に管理されており、期日内の対策検討および、深刻な場合のユーザへの通知などを徹底している場合等が該当する。ただし、他の製品の公開情報を積極的に利用するという点では弱い部分もあり、他の製品では解決済みで公開されている脆弱性が、自社製品において発見される場合もある。

- レベル４

組織の方針として脆弱性対策方針を定めると共に、一般への公開を行なっている水準である。発見された脆弱性関連情報をJPCERT/CC等の機関に報告し、また積極的に社外の脆弱性関連情報を活用する場合等が該当する。脆弱性関連情報を登録することで、オープンな検証ツール上で当該の脆弱性検証が可能になったり、オープンソースのドライバやライブラリに対策が施されたりする等の効果がある。この水準では、既知の脆弱性の作り込みを避けることが可能になることから、組込みシステムのセキュリティの底上げが期待でき、本書における一つの目標となる。

3.3.4.　廃棄方針

「廃棄方針」は、製品を廃棄する際の、機密情報の消去等に関わる取組み方針である。ここでは、廃棄方法の有無が主な基準となる。廃棄に関する各レベルの取組み内容をまとめると以下のようになる。

- レベル１

廃棄方法が考慮されていない水準である。製品を廃棄するか使い続けるかは顧客の利用方法に依存するため、仕様書にも特に明記がない場合等が該当する。

- レベル２

廃棄方法が仕様書等に明記されている水準である。製品には寿命があることから、廃棄時のセキュリティリスクの通知と推奨の対策を明記することは必須と考えている場合等が該当する。ただし、製品によっては機密情報の消去に専用のツールが必要になる場合もあり、実質的に対策が困難な場合もある。

- レベル３

廃棄時のセキュリティリスクの軽減方法が用意されている水準である。データ破壊ツールや回収システムを整備している場合等が該当する。ただし、これらの手段を無償で提供する必要はない。

- レベル４

客観的な基準に基づいたセキュリティリスクの軽減方法が用意されている水準である。ストレージを有する機器で米国の国家安全保障局（NSA : National Security Agency）等に基づく消去方法を実施している場合等が該当する。トラッキング技術を利用した、廃棄された機器のシステムへの再接続を禁止する運用基準等もこれに該当する。この水準では、ユーザが廃棄する製品にプライバシーや機密情報が残らないことを納得できるという点で、本書における一つの目標となる水準である。

4. 各レベルの詳細

4.1. マネジメントにおける取組み

4.1.1. セキュリティルール

組込みシステムを提供していく上で、開発元のセキュリティに対する取組みは、場当たり的になってはならない。このため、セキュリティに対する取組み方針を定め、具体的に行うべきこと、および行ってはならないことを組織内の規約（セキュリティルール）として明確化する必要がある。

例えば、組込みシステムを構成するモジュールを外部から調達する際には、セキュリティ上の問題を起こさないように調達に関するセキュリティルールが必要である。内部で開発するモジュールに関しては、セキュリティ面における技術的な対策や機能の実装方法や、開発および製造におけるプロセスの信頼性を保障できる開発体制・環境を実現するセキュリティルールが必要である。また、ユーザサポート等を行う上で、ユーザの個人情報を適切に取り扱うためには、組織のセキュリティマネジメント体制を実現するセキュリティルールが必要である。具体的なセキュリティルールとしては以下のようなものが考えられる。

- 情報セキュリティ基本方針
  組織全体の情報セキュリティに対する取組みの方向性を、事業上の要求事項、および関連する法令および規則に従って規定する。

- 組織全体に関する情報セキュリティ管理規則
  情報セキュリティの責任者の割当てや、セキュリティを維持するための組織作りに関する規則を規定する。

- 人的リソースに関する規則
  ユーザが組込みシステムを安全に使用する上で機密扱いとしておかなければいけない情報の漏えいを防ぐため、適切な人員を雇用するための規則や機密保持契約、および、セキュリティ教育に関する規則を規定する。これらの規則は、派遣や請負により組込みシステムの開発に関与する人員に対しても適用することが必要である。

- 開発体制・環境に関する規則
  組込みシステムの開発環境に対する、承認されていない物理的アクセスやウィルス等の混入を防ぐための規則を規定する。

- 設計に関する規則
  組込みシステムの設計・実装において、セキュリティ事象の予防や、検知・回復を行う上で必要なセキュリティ機能を選択するために必要な規則を規定する。

- 調達に関する規則
  外部調達するハードウェアやソフトウェアが、必要なセキュリティ機能を正しく実装していることを第三者から見ても保証されるようにするために必要な規則を規定する。

・ 運用に関する規則

ユーザの利用中に組込みシステムの脆弱性が発覚した場合、対策版のソフトウェアを開発すると共に、ユーザにそのソフトウェアを適用してもらえるよう連絡する必要がある。まず、問題が発生した場合の処理手順や、連絡に必要なユーザの個人情報などを適切に取り扱うための規則を規定する。

レベル	取組み概要
レベル1	セキュリティルールの策定は行っていない
レベル2	セキュリティルールは開発者が自主的に策定している セキュリティルールは開発責任者や担当者主導で策定される
レベル3	組織としてセキュリティルールを策定し、規則集として明文化している
レベル4	組織としてセキュリティルールを策定し、規則集として明文化しており、担当者が規則を順守していることを確認するための手段も規定している

4.1.2. セキュリティ教育

近年組込みシステムは、高度なハードウェアと大規模なソフトウェアを組み合わせて実装されるようになりつつある。この結果、本来できないはずの操作や、権限を逸脱した情報へのアクセスを可能にするといった欠陥（脆弱性）が、組込みシステムに入り込む可能性も高まっている。これらの脆弱性は、自社もしくは外部委託により開発されたソースコードに作り込まれる場合や、ベースとなるハードウェアやソフトウェアに含まれている場合がある。

脆弱性を攻撃する方法は、研究者や攻撃者によって日々研究されており、新しい脆弱性も日々発見されている。このため、情報セキュリティに関する知識を学習することにより、以下に関する知識を習得した開発者を育成することが望ましい。

- 現時点で知られている脆弱性に関する知識

  開発したソフトウェアで発見される脆弱性は、既存のソフトウェアの事例として既に報告されていることが多い。例えば、JVNの脆弱性対策情報[4] では、組込みシステムの管理用GUIにおけるクロスサイト・スクリプティングおよび、クロスサイト・リクエスト・フォージェリの問題や、ディレクトリ・トラバーサル、SQLインジェクション等の様々な脆弱性の事例が掲載されている。このため、インターネット等で公開されている具体的な事例について学び、仕様設計やコーディングを行う際に同様の脆弱性を作り込まないようにすることが必要である。

- セキュア・プログラミングに関する知識

  組込みシステムを実装する際にセキュリティホールが作り込まれることがある。例えば、認証や暗号機能を提供するライブラリの使い方が誤っていると、認証処理をバイパスして組込みシステムを操作されたり、暗号化したデータを解読されたりすることがある。また、プログラムで想定していない長さや文字コードのデータを外部から組込みシステムに入力することにより、誤動作が引き起こされることもある。セキュアなプログラミングを行うためのヒントについては、セキュア・プログラミング講座[5] や、e-Learningサイト、書籍等で紹介されている。

- セキュリティテストに関する知識

  設計工程および実装工程において、セキュリティ対策が正しく行われたことをテストすることが必要である。具体的には、（１）設計ドキュメントをレビューし、考慮すべきセキュリティ対策に漏れがないか検証することと、（２）記述したソースコードに脆弱性がないことを検証することと、（３）開発した組込みシステムを動作させながら脆弱性の有無を検証することが必要である。セキュリティテストに関するヒントも、e-Learningサイトや書籍等で紹介されている。

  ソースコードのセキュリティ検証には、専用のソースコード解析ツールがある。特にネットワークに接続するタイプの組込みシステムに対しては、TCP/IPを実装したソフトウェアの既知の脆弱性を

---

[4] JVN脆弱性対策情報（http://www.ipa.go.jp/security/vuln/documents/index.html）

[5] 新版「セキュア・プログラミング講座」のねらい
（http://www.ipa.go.jp/security/awareness/vendor/programmingv2/index.html）

体系的に検証するツール[6] も提供されており、これらのツールに関する知識も深めていくことが望ましい。

レベル	取組み概要
レベル1	セキュリティ教育は実施していない
レベル2	セキュリティに関する分科会（小集団活動）が自主的に活動している 必要に応じて開発責任者や開発者主導で知識を習得したり、勉強会が行われたりする
レベル3	セキュリティ教育が業務に組み込まれている セキュリティに関する講習会や研修の受講が奨励される
レベル4	セキュリティ教育が業務に組み込まれている 専門家による研修カリキュラムが義務付けられている

---Coffee Break---

脆弱性に関連する用語は、海外で定義された用語が圧倒的に多い。そして用語の分かり難さに加え、脆弱性が存在する範囲の広さにも圧倒される。しかし、それを理解しなければいけない状況になってしまった。なぜならば、今までは「限られた利用環境の中で、限られた資源を使い、限られた機能」を実現すればよかった。しかし、ネットワークに接続され、これまでよりも便利かつ高度で大規模な組込みシステムを実装しようとすればするほど､我々自身が開発した以外のものも利用せざるを得ない。つまり、我々が完全に制御できないものの脆弱性とも付き合っていかざるを得ないからだ。

脆弱性が内在し、それが利用できる状態ならば、我々が開発した製品もまたターゲットの１つになっている。たとえそれが特定の用途向けに専用化されていたとしても、「この製品には機密情報は含まれていないから大丈夫」ということには、残念ながらならない。攻撃者は、その製品自体に興味はなくても、その製品に内在する脆弱性を利用し、処理能力を少し借りて、他のターゲットを攻撃する踏み台にもできるからだ。つまり、攻撃者の手伝いをする可能性がある製品となるわけだ。

有名なセキュリティシステムを導入しても、全ての問題が片付くわけではない。TPM（Trusted Platform Module）を例にとってみよう。TPMはソフトウェアの改ざんを検知できたり、その機器が正当であることを確認できたり、暗号処理ができたりと、至れり尽くせりだ。しかし、TPMもシステムの1つに過ぎない。BLACK HAT 2007というセキュリティ関連のカンファレンスで「TPMkit: Breaking the Legend of Trusted Computing（TC [TPM]）and Vista（BitLocker）」というプレゼンテーションがある予定だった。しかし発表は中止された。理由はなぜか不明だ。そして、非常に興味深いツールであるにも関わらず、Googleで日本語サイトの検索をすると、たった4件しかhitしない。これはどういうことなのか。常に疑問の目を持つことが大切だ。

まずやるべきことは、我々の製品に何が使われているのか理解することだ。そして使われているものの脆弱性と悪用例にどのようなものがあるのか、日本に限らず調査しよう。まずはそこからだ。予想以上に悪用される可能性があることが分かるだろう。しかし、調査の段階では分からないことだらけで、本当に我々の製品に影響があるものなのか判断もできないだろう。

だからこそ、知識が必要なのだ。

[6] TCP/IP脆弱性検証ツール（http://www.ipa.go.jp/security/vuln/vuln_TCPIP_Check.html）

4.1.3. セキュリティ情報の収集

組込みシステムをユーザに提供していく上で、ライフサイクル全体にわたり、組込みシステムのセキュリティに関わる情報（セキュリティ情報）を収集し、活用していくことが必要である。開発フェーズにおいては、組込みシステムのベースとなるハードウェアおよびソフトウェアにセキュリティ上の問題がないことを確認すべきである。開発者が収集すべきセキュリティ情報には以下のものがある。

・ 脆弱性に関する情報

例えばLinux等の日本国内で広く利用されているソフトウェアを用いて組込みシステムを開発している場合、脆弱性対策情報データベース「JVN iPedia」[7] を検索することにより、今までに発見された脆弱性に関する情報を収集することができる。さらに、JVN iPediaに登録された多数の情報の中からユーザが、ユーザ自身に関係する情報のみを効率的に収集できるように、フィルタリング条件設定機能、 自動再検索機能、脆弱性対策チェックリスト機能などを有した脆弱性対策情報収集ツールとして「MyJVN」[8] がある。

日本国内でコンピュータセキュリティの情報を収集し、インシデント対応の支援、コンピュータセキュリティ関連情報の発信などを行っているJPCERT/CCでは、脆弱性関連情報の悪用、または障害を引き起こす可能性を最小限に食い止めるため、一般公表前の脆弱性関連情報を、情報システムの開発者等に必要に応じて公開する「脆弱性情報ハンドリング」[9] と呼ぶプロセスを実行している。利用するソフトウェアに関する脆弱性関連情報をいち早く入手するには、この脆弱性情報ハンドリングに参加するとよい。

Webニュースサイトでも各種のIT製品で発生したセキュリティ問題に関する記事が取り上げられている。例えばSecurityFocus[10] では、ソフトウェア製品に検出された脆弱性、攻撃手口と防御手法、セキュリティ・ツール等を収集、掲載している。

・ 認証や情報保護に使用する暗号技術に関する規格動向

暗号/認証技術は、世界中の研究者やクラッカーにより、攻撃方法が日々研究されている。このため国内では、客観的な評価により安全性および実装に優れると判断された暗号技術をリスト化する暗号技術評価プロジェクトとして、CRYPTREC（Cryptography Research and Evaluation Committees）が活動している。認証や情報保護に暗号技術を利用している組込みシステムを開発する際には、CRYPTRECが提供している報告書[11] を参照すべきである。

レベル	取組み概要
レベル1	セキュリティ情報は特に収集していない
レベル2	開発責任者や開発者主導で必要に応じて情報収集を行っている
レベル3	組織として情報収集を行い、収集結果は開発者が参照可能にしている
レベル4	脆弱性情報ハンドリングに積極的に関与し、脆弱性関連情報の活用に努めている

---

7) JVN iPediaホームページ（http://jvndb.jvn.jp/index.html）

8) MyJVNホームページ（http://jvndb.jvn.jp/apis/myjvn/）

9) 脆弱性情報ハンドリングホームページ（http://www.jpcert.or.jp/vh/）

10) SecurityFocusホームページ（http://www.securityfocus.com/）

11) CRYPTRECホームページ（http://www.cryptrec.go.jp/report.html）

4.2.　企画フェーズにおける取組み

4.2.1.　予算

組込みシステムのライフサイクル（企画・開発・運用・廃棄）の全てのフェーズにおいて予算の確保が必要になる。特に実績のあるセキュリティ関連のソフトウェア（OS、暗号化ライブラリ等）は、開発時の初期費用だけでなく、アップデート費用が発生するため、継続した予算確保が必要となる。

また、セキュリティ上の問題は事前に予測することが困難であることから、販売後しばらく経ってから問題が発生する場合が多々あるため、開発時のセキュリティ予算の確保とともに、リスク回避の観点から全社的かつ継続的なセキュリティ予算の確保が望まれる。経営層も含めたセキュリティ意識を全社的に徹底させる上で、開発計画に基づいた稟議・決済に関連する書類には、以下のセキュリティ予算項目を載せることが望ましい。

- 導入時
  - 調査費（書籍・セミナー・研修費用など）
  - 購入費（ハードウェア・ソフトウェア・本番およびテスト用の証明書など）
  - 購入物品管理費（リビジョン管理システム構築費など）
  - テスト環境構築費

- 本番運用時
  - 調査費（製品関連ハードウェア・ソフトウェア・証明書のアップデート確認費用など）
  - テスト環境・ハードウェア・ソフトウェア・証明書の保守・更新費
  - リビジョン管理システム[12] 運用費
  - セキュリティ上の問題が発生した場合を想定しての対策費用（製品単体ではなく、全製造製品での確保でも可）

レベル	取組み概要
レベル1	セキュリティに係る予算は用意されていない
レベル2	プロジェクトリーダー（開発責任者）の要求があった場合に、予算確保が容認される 開発責任者や開発者から要求があった場合に、検討される
レベル3	開発プロセスの一つとしてセキュリティ確保のために一定の基準で予算を割り振ることが前提として組まれている
レベル4	組織にセキュリティ部門等を設置するための予算を確保し、活動する

[12] 購入したソフトウェア・ドライバが、製造物に適切に適用され続けるように、そのリビジョン（バージョン）を管理するシステム。

4.3. 開発フェーズにおける取組み

4.3.1. 設計

一般的な組込みシステムの開発プロセスでは、その初期段階で機能仕様や性能などを決定する設計工程がある。この段階で機能仕様を実現するためのハードウェアとソフトウェアについて構成が決定され、その後各機能について具体的な検討が行われる。実際の開発現場においては、本書で述べるようなセキュリティ対策が意識されることは少ないのが実情ではないだろうか。

しかし、近年はネットワークに接続する機能を備えた組込みシステムが増加していることから、これらの機器が攻撃者に狙われる危険性があることを認識しなければならない。万が一、攻撃に対する弱点を持ったまま市場に出荷され、それが不具合として問題化すると、その回収修理に膨大な費用を要してしまう。数年前に報告された携帯電話の不具合問題では、その回収費用が100億円以上に達したとされている。こうしたことから、いわゆる開発の上流にあたる設計工程において、該当の組込みシステムが考慮すべきセキュリティ要件を抽出し、どのような対策を実施するかを検証することが重要になってくる。

組込みシステムの設計工程で検討すべきセキュリティ要件の例について下記に挙げた。これらの中には、利用形態によってはセキュリティ要件として位置づけられないものもあるかもしれない。そのような場合は、設計する機器の用途や想定されるユーザ層に応じて取捨選択する。

・ 機密情報の流出防止

ネットワークに接続された組込みシステムが攻撃を受けた場合に、内部に蓄積された個人情報などの機密情報の流出を防止することが必要である。機密情報にはユーザの不利益となるような情報資産も含まれる。（※情報資産については、別項「表 3-2 組込みシステムで保護すべき情報資産」を参照のこと。）情報資産の機密度に応じて、データの論理的あるいは物理的な配置、暗号化方法、暗号強度について仕様を決定する。

対策：

- 機密度に応じた暗号アルゴリズムの仕様を策定する。
- 暗号化メモリ、暗号化ハードディスクなどの採用を検討する。
- USBメモリやICカードといった外部メモリの装着を可能とし、組込みシステム内部の不揮発性メモリに機密情報を持たせないような構成を採用する。
- アクセス制御として、本人認証を強化するために、パスワード認証やバイオメトリック認証などを採用する。

・ 障害復旧

攻撃を受けたことで生じるフリーズや再起動の繰り返し等、組込みシステム単体の障害を防止する。

対策：

- 障害を検出するための自己診断、自己回復、電源断機能を実装する。
- 障害を機器のユーザへ通知する機能、またはオンラインで通知する機能を実装する。

・ 踏み台攻撃への対策

ネットワークに接続された組込みシステムが攻撃を受け、これを踏み台としてさらに他の機器へ攻撃を行うなどの障害を防止する。また、ネットワークに接続された組込みシステムの異常動作を誘発し、ネットワーク上のトラフィックを極端に増大させることにより他の機器への影響を及ぼすような障害を防止する。

対策：

- ネットワークを介した攻撃を検出し、攻撃を無視する仕組みを検討する。ネットワークを介した攻撃を検出する方法の例を以下に示す。
  - 特定期間における多すぎるパケットデータの受信
  - 特定期間における多すぎるパスワード誤入力
  - 想定外に大きなサイズのパケットデータの受信
  - 使用していないポートへのアクセス

・ アラート機能

組込みシステムが攻撃を受けた場合、そのことを機器のユーザへ警告する方法について仕様を検討する。また、ユーザが誤操作や誤設定を行った場合に、機器のユーザへ通知する方法について仕様を検討する。

対策：

- 攻撃を受けたことを機器のユーザへ警告するユーザインタフェースを実装する。
- 警告メッセージは、ネットワーク切断を促す等の緊急避難方法を明示する。
- 取扱説明書やその他ユーザガイド類において、警告メッセージの詳細な解説を明示する。
- 機器の設定を変更する際にパスワードによる制限を設ける。
- ポート番号などの変更方法や使用制限について検討する。
- 機器のユーザがネットワークに関する危険な設定を行った場合に、警告メッセージを表示するユーザインタフェースを実装する。

・ ロギング機能

ネットワーク経由で攻撃を受けたことや、その他の攻撃を受けたことをログとして組込みシステム内部に記録する機能を検討する。

対策：

- 送受信ログの記録機能の実装是非を決定する。記録内容は、送受信双方のIPアドレス、ポート番号、データサイズ、時刻等である。
- 機器のユーザによる認証のログの記録機能の実装是非を決定する。記録内容は、ユーザID、ログイン回数、ログインエラー回数、時刻等である。
- 各種ログの保存期間を検討し、データサイズやメモリ構成を決定する。
- ログをオンラインで管理する場合は、(暗号化を含む) 通信規約やフォーマット等を決定する。

・ サービス機能

工場における検査機能、サービスマン用のメンテナンス機能、販売時の店頭デモ機能等、機器のユ

ーザが使用しない特殊機能へのアクセス制限について検討する。また、開発用のデバッグコネクタ、サービスマン用の通信端子などからの不正なアクセス防止対策を検討する。

対策：

- 各種特殊機能へのアクセス方法を一定の手続きを必要とするものとし、機器のユーザに容易に推測されないようにする。
- 出荷後に不要となる機能は削除する。
- 各種特殊機能へのアクセス方法の情報が、インターネット上に流出していないかチェックする
- 量産用のハードウェアではデバッグコネクタを実装しない、場合によっては配線パターンを削除するなどの対策を行う。
- サービスマン用の通信端子は、コネクタ形状や実装方法を工夫し、容易なアクセスを防ぐ。
- サービスマン用の通信端子は、組込みシステム内部の情報の読み出しや書換えに用いられる場合がある。このような場合は、マイコン側と通信するためのフォーマットを一定の手続きを要する形式とするなどの工夫をする。

・ 組込みシステムのハードウェアに対する直接的な攻撃対策

筐体の不正なこじ開け、基板上のデバイスへの不正なアクセス、プロービング、不揮発性メモリのデータの書換え、等に対する対策を検討する。

対策：

- 不正なこじ開けやハードディスク、メモリなどの取り外しを検知した場合に、機密情報を消去する仕組みを実装する。
- 不正なこじ開けや取り外しを検知した場合に、オンラインで通知する仕組みを実装する。
- 機密情報を扱うデバイスには、BGA（ボールグリッドアレー）パッケージのデバイスを選択するなど、形状を工夫することでプロービングを防止する。
- デバイス実装後の基板上に物理的または化学的なマスク処理を行い、プロービングを防止する。

・ 廃棄機能

機器の譲渡や廃棄時において、個人情報等の情報資産を初期化または破棄するための機能または手順を検討する。

対策：

- 機密度の高いデータを他のデータと分けて格納し、部分的に消去を可能とするためのハードウェア構成の検討
- 機器のユーザが消去や初期化の作業を、容易に実行できるユーザインタフェースを実装する。
- ハードディスクへの記録であれば、ディスク上のセクタを物理的に消去する仕組みを検討する。
- USBメモリやICカードといった外部メモリの装着を可能とし、組込みシステム内部の不揮発性メモリに機密情報を持たせないような構成を採用する。

・ サイドチャネル攻撃への対策

近年、暗号アルゴリズムを実装したときに、その実装環境において物理的デバイスから漏えいしたり、不正な行為を行って得られる物理量（例えば処理時間や消費電流等）やエラーメッセージを利

用したりして攻撃対象の機密情報を得る、「サイドチャネル攻撃」と呼ばれる攻撃手法の存在が明らかになってきた。代表的なサイドチャネル攻撃の例としては次のようなものがある。

表 4-1 代表的なサイドチャネル攻撃の例

攻撃手法	概要
プローブ攻撃	LSIのパッケージを開封し、チップ表面のデータバスに探針(プローブ)を当て、回路を動作させながらバス上のビットの変化を観察し、解析する
電力解析攻撃	タイミング攻撃と同様に、チップが消費する電力を解析することにより、内部の機密情報を推定できることがある。単純電力解析(SPA)や、電力差分解析(DPA)といった、様々なバリエーションが検討されている
電磁波解析攻撃	LSIの周辺にコイルを近づけ、電磁波放射を測定して動作を推測する手法である。得られた電磁放射量は、電力消費量と同様な解析が可能である
故障利用攻撃	人為的にモジュールに故障を起こさせた状態で処理を行わせ、その結果から機密情報を推定する手法である。故障を引き起こすには、定格外電圧や、定格外のクロック周波数を印加する方法や、電磁波・放射線・強力な光を照射する方法がある
タイミング攻撃	機密情報を処理する実行時間を測定することにより、その機密情報を推定する方法である。例えば、モンゴメリ乗算を行う際に、演算の最後の段階で中間結果の値によって、時間がかかる剰余算を行う場合と、行わない場合に分けられる。このため、入力データを変更しながらモンゴメリ乗算を行い、その実行時間を測定することで、内部の機密情報が推定できることがある

対策：

各攻撃手法に対する対策手法を以下にまとめる。

- プローブ攻撃を防ぐため、機密情報を処理するLSIは外部から動作を観測することを妨げるような保護層を備えたものを使用する。また、回路を取り囲むように金属層を設け、そこに流れる電流をモニタした上で異常を検出した場合には機密情報を破壊する方法や、保護層を取り除こうとするとチップ自体が破壊されるようにする（耐タンパメカニズム）方法もある。
- 電力解析攻撃、電磁波解析攻撃を防ぐため、タイミング攻撃の対策と同様に、演算内容により差異が生じないようにする。
- 故障利用攻撃を防ぐため、故障の状態を検知する仕組みを設け、故障を検出した場合には機密情報を破壊する。
- タイミング攻撃を防ぐため、演算アルゴリズムにダミールーチンを挿入し、演算内容により差異が生じないようにする。

レベル	取組み概要
レベル1	設計段階においてセキュリティ対策はほとんど行われず、仕様書類にも記述されない
レベル2	設計段階で行うべきセキュリティ対策は開発責任者や開発者の判断で実施される 組織としてのルールはなく、実施内容の監査についても明確になっていない
レベル3	組織として設計段階で行うべきセキュリティ対策のルールが制定されている それに基づいた設計プロセスが実施される。実施内容の監査の仕組みはない
レベル4	組織として設計段階で行うべきセキュリティ対策のルールが制定されている それに基いた設計プロセスが実施され、その内容をセキュリティ部門が監査する仕組みが機能している

---Coffee Break---

例えば電力差分解析（DPA）攻撃対策の研究事例としては、どのような暗号演算を実行しても消費電力が一定となるような回路構成やアルゴリズムの改良によって機密情報と消費電力の相関をなくすといったものがある。しかしながら、学会などで発表される攻撃対策法には、コストがかかりすぎたり、現

実のLSI設計においては非現実的な場合がある。量産レベルのシステムLSI設計において、確立された方法論はないのが現状である。実際のシステムLSI設計で明示的に電力解析攻撃対策をとらなかったとしても、例えば微細プロセスのセルを採用することにより、消費電力がごく微小なものとなる。このことは結果的に電力解析攻撃を難しくすることにつながる。
また、暗号演算をCPUで行うと電力解析されやすいため、専用回路に分けて設計したり、暗号鍵を頻繁に変更したりするといったロジック上の対策も、電力解析攻撃には有効な対策となる。
ここで重要なのは、暗号演算などを行うシステムを扱う場合、サイドチャネル攻撃を過度に警戒して不要なコスト増をまねくのは本末転倒であるが、決して軽視せずに対策の有効性と現実の設計のバランスを考慮した設計を行うことである。

4.3.2. 開発プラットフォーム選定

開発プラットフォームとは、開発する製品を構成するハードウェアや、母体となるソフトウェア等を指すものとする。組込みシステムを開発する場合、目的とする機能や性能を実現するために、CPUやメモリといったハードウェア資源を選定し、それらを制御するためのOSや各種ソフトウェアを実装していく。基本的な開発スタイルは、全ての構成要素をゼロから選定し、独自に開発用のプラットフォームを構築する方法である。

しかし、近年では開発期間の短縮とコストダウンの観点から、CPUや各種デバイスの供給元である半導体メーカが推奨する「リファレンスボード（基板）」等をベースとし、そこに自社のカスタマイズを加えていく開発スタイルも多く見られる。こうしたリファレンスボードは、開発時におけるデバッグ作業を容易にしたり、拡張性を念頭に設計されたりしており、セキュリティ面は考慮されていない場合が多い。ハードウェア設計の際に、動作の安定性を求めてデバイス選定や回路設計をそのまま流用するといったケースが考えられるが、これはセキュリティ面から考えると非常に危険である。具体的な脅威としては、基板上のデータバス上に機密情報が非暗号化された状態で流れ、プロービング等によって攻撃者に盗み取られるといった物理的な攻撃事例が報告されている。リファレンスボードなどを参考に設計を行う場合は、こうした攻撃者による物理的な攻撃に対する対策について十分な検討を行うことが重要である。

ハードウェアの選定においては、攻撃者による物理的な攻撃への対策を考慮する必要があり、次のような点に注意する。

- 基板のデータバス等に機密情報が流れ、それらが攻撃者によって容易に測定できないような回路構成とすること。リファレンスボードなどを参考にシステム構成を決定する場合は、回路構成をそのまま流用可能か考慮すること。
- 機密情報を扱うデバイスには、BGA（ボールグリッドアレー）パッケージのデバイスを選択するなど、形状を工夫することでプロービングを防止する。

次にソフトウェアについて考える。組込みシステムのソフトウェアは、ハードウェア構成に基づいて決定される。CPUやデバイスに合わせてリアルタイムOSや制御ドライバソフトを選定する。OSとドライバの他に、要求される機能を実現するためのソフトウェア群がパッケージ化された製品も販売されている。また、近年は製品の多機能化と大容量化に伴って、例えば組込Linuxのようなオープンソースソフトウェアを採用する動きも盛んである。

いずれの場合においても、採用する組込みソフトウェアパッケージ販売元のセキュリティ対策に関する姿勢について事前に確認することが重要である。ここでいう姿勢とは通常のバグ修正や機能改善に加え、脆弱性関連情報の収集やその対策が、修正プログラムのリリース等によって実施されているかといったサポート体制の充実度を意味するものである。

具体的な脅威の情報については、脆弱性対策情報データベース「JVN iPedia」[13)]から得ることが可能である。これは本書の「4.1.3 セキュリティ情報の収集」項に述べる通りである。組込みソフトウェアパッケージ販売元が、これらの脆弱性関連情報にどのように対応しているか等を見極めることも導入時

---

13) JVN iPediaホームページ（http://jvndb.jvn.jp/index.html）

の１つの指標となる。あるいはそのパッケージの他社での採用事例などから、脅威に対する強度を見極めることもできる。

組込みソフトウェアパッケージなどを利用せず、独自のプラットフォームを構成する場合は、前述の脆弱性関連情報には十分注意を払う必要がある。常に動向を監視し、然るべきタイミングで修正プログラム等の提供を行っていくことが重要である。

ソフトウェアの選定においては、次のような点に注意する。

- 組込みソフトウェアパッケージを利用する場合、販売元の脆弱性対策に関するサポート体制について事前に確認を行う。
- 独自の開発プラットフォームを構成する場合、脆弱性対策情報を自ら収集する必要がある。その対策についても自主的に実施しなければならない。

レベル	取組み概要
レベル1	コスト優先、納期優先で選定され、セキュリティについてはほとんど考慮されない
レベル2	開発時に判明している脆弱性への対策は実施する 出荷後は特に実施されない 開発プラットフォームの選定を実施する担当者または部門がない その製品の開発責任者や開発者によって脆弱性対策の意識が異なり、組織として統一した取組みはなされない
レベル3	組込みソフトウェアパッケージを利用する場合、販売元からの脆弱性対策プログラム等のリリース情報を監視している 出荷後の製品に対してもソフトウェアアップデート等によって対策を実施している 同じ製品を扱う部門内では脆弱性対策への意識が高く、統一した取組みがなされている
レベル4	全社的に脆弱性対策を実施している 出荷後の製品に対してもソフトウェアアップデート等によって対策を実施している 販売元の脆弱性対策に関するサポート体制について事前に確認を行う デバイスや回路構成においてセキュリティを考慮した選定を行っている 製品のセキュリティを高める目的で、独自のプラットフォームの開発や調達先への改善要求活動などを実施している

4.3.3. ソフトウェア実装

ソフトウェアに発生するセキュリティ上の問題は、一般的なバグのように運用時に自然発生するものに加え、直接的または間接的攻撃により、組込みシステム特有の振る舞いが解析され、その情報を足掛かりに脆弱性へと発展するものがある。そのため、セキュアなソフトウェア実装においては、直接的および間接的攻撃への対策のために、ソフトウェア耐タンパ実装およびセキュアコーディングを考慮することを推奨する。前者はメモリ内のデータやプログラム自身、バス上に流れるデータ等を対象に、組込みシステム自身を直接解析する攻撃からソフトウェアやデータを保護するための技術であり、ソフトウェア難読化やバス暗号化等の対策が挙げられる。後者は不正なパケットや命令コードを送付し、その振る舞いから組込みシステムを間接的に解析する攻撃からソフトウェアやデータを保護するための技術であり、パケットフィルタリングやバッファオーバフロー対策等が挙げられる。

セキュアなソフトウェアを実現するにあたっては、このような取組みが性能やデバック効率とトレードオフの関係にあることに注意すべきである。

ソフトウェアの実装段階における作業は、大きく分けてソースコーディングと、そのレビューという形で進められる。製品の品質を向上させることを目的として、様々なコーディング技法やレビュー手法が実践されている。開発するソフトウェアの規模によって状況は異なるため、ここで画一的にこれらの手法を挙げることは難しい。ここではソフトウェアの実装段階における脆弱性対策について、既存の研究報告例を紹介する。自らの属する組織が後述する取組みレベルのどこに属するかを見極めた上で、実態に即した運用を実践していただきたい。同時に、コーディング技法やレビュー手法の習得にあたっては、専門的な知識が必要となる。セキュリティ教育等を通じた組織的な取組みが求められる。

研究報告例１：
バッファオーバフローやフォーマット文字列攻撃に関する対策として、IPAでとりまとめを行っている、「セキュア・プログラミング講座[14)]」が参考となる。本講座は組込みシステムに特化した構成にはなっていないが、特に「C/C++言語編」は組込みシステムのソフトウェア開発において価値のある内容となっている。

研究報告例２：
IPAの研究報告、「ソフトウェアのソースコードを読解・批評する方法論の確立を目指して[15)]」を参考にする。あくまで作業中の内容ではあるが、暫定版として公開されている。ソースコードを読解し、評価およびレビューする手法やテクニックについて述べている。ソフトウェア言語やOSを単位として主査を配し、ソースコードの読解に主眼をおいている点が特徴である。

レベル	取組み概要
レベル1	セキュリティを意識したコーディングはなされていない

[14)] 新版の「セキュア・プログラミング講座」のねらい
（http://www.ipa.go.jp/security/awareness/vendor/programmingv2/index.html）

[15)] ソフトウェアのソースコードを読解・批評する方法論の確立を目指して
（http://www.ipa.go.jp/security/fy17/lab/codeblog.html）

	ソースコードレビューは実施されない または、実施されてもセキュリティに関する検証はなされていない
レベル2	脆弱性対策を目的としたコーディングルールやレビューの手法については、担当者あるいは一部グループなどによる任意的な活動に依存する 同じ部門内でも統一した活動がなされているわけではない それらの内容についても、開発責任者や開発者の知識と経験に基づいて実施されるため、明確な判断基準はない
レベル3	脆弱性対策を目的として、コーディングやレビューを始めとする実装段階におけるルールが組織的に規定されている 担当者は、このルールに則って作業を進める これらの結果について部門内または部門間で監査する業務フローが規定されている
レベル4	レベル3の取組みに加えて、セキュリティを専門とする部門が存在し、コーディングやレビューにおいて指導を行う また、この組織がルール策定に関与し、不具合とその対策などの情報を基にしてルールの改定等を続けている

---Coffee Break---

要件を満足していながら、セキュリティ上の問題があるコードを作成してしまうのはどういった場合だろうか。ここではあるデータを暗号化する関数を持つプログラムを例に考えていく。
データ保護の観点から、暗号化をサポートしたソフトウェアは非常に多くなっている。しかし攻撃者の観点からすると、暗号化したいと思うデータは価値があるデータであるという特性から、暗号化するタイミングのメモリアクセスを解析することで、何百MBというデータの中からでさえ重要データを探し出すことが可能になる。またその時にデータの暗号化鍵すら分かってしまう可能性もある。
では攻撃者はどうやって暗号エンジンの場所を探し出すのだろうか。一つの方法として、AESやDESが、特定の処理を規程回数繰り返すという特性を利用し、ループ処理に着目して解析箇所を絞り込むという手法がある。もちろんこれにより一意に暗号エンジンの場所を特定することはできないが、解析に掛かる時間は短くなる。
このような攻撃により重要な情報が漏えいした場合、それはバグと呼べるだろうか。そしてこの問題を解決する手段は存在するだろうか？
残念ながら既存の技術では、解析を困難にする対策しか行えない。組込みシステムにどの程度の解析困難さを求めるかは、守るべき情報資産の価値と許容可能な製造コストに基づき、独自に判断するしかないのが実状である。

4.3.4. 開発の外部委託における取組み

組込みシステムの開発においては、大規模化と開発コスト抑制などの観点から、社外へ開発を委託する場合がある。基本的な脆弱性対策の取組みについてはここまで述べてきた通りであり、委託先にも同様の取組みを求める必要がある。しかし、実際には双方の風土や文化の違いにより、同じ組織内の活動をそのまま適用することは難しいであろう。さらに、多人数による大規模なソフトウェア開発では、地理的な制約などによるコミュニケーション障壁についても考慮する必要がある。

特にソフトウェア開発を外部委託する場合は、開発プロセスが外部から見えにくくなることによって脆弱性が作り込まれ易い。コーディングやレビュー等の実装ルールについて意思徹底が行き渡らず、ミスが見過ごされる危険性がある。また、委託先の都合によってスキルが十分ではないエンジニアが設計を行うことも考慮しなければならない。

外部委託の形態は一様ではない。ハードウェアは内製としソフトウェアのみを委託する場合、ソフトウェアを部分的に委託する等、製品構成や開発コストの都合によって様々なパターンが考えられる。基本的には、本書の開発フェーズにおける取組みの各項の内容を実践することが脆弱性対策としては有効であり、自部門の外部委託形態に即して取り入れていくことが必要である。

委託先が様々な形で開発に関与するため、内部で規定されたセキュリティルールに従って委託先にも取組みを求めていく必要があり、そのポイントを挙げてみたい。

- ハードウェアやソフトウェアの土台となる部分の設計に発注先が関与する場合は、設計ルールや選定基準について発注託先へ明示し、丸投げ状態にならないよう役割分担を決定する。
- ソフトウェアの実装部分を委託する場合は、コーディングルールやレビュー手法を事前に明確化し、見積り条件として提示する。
- 取組みが要求通り実施されていることを監査するために、合同レビュー会を開催し、設計ドキュメントの提出を求める。それらを監査し、改善要求を行えるように体制を整備する。
- セキュリティ評価およびデバッグの手法について事前に検討し、検証ツールの購入費用や人員確保のコストについて双方で相談する。
- セキュリティ対策に関連する検収条件を規定し、セキュリティに特化した成果物を受領する。
- 委託先とのコミュニケーションをとるための手段や環境について検討し、実態に即したセキュリティルールの運用が行えるように配慮する。
- セキュリティ上の問題が発生した際の責任範囲を、契約上で明確化する。

レベル	取組み概要
レベル1	開発を外部委託する場合に、脆弱性対策に関する取組みについては考慮されない セキュリティに関する機能が含まれていても、通常の開発委託と同様の扱いとしている
レベル2	開発を外部委託する場合の脆弱性対策については、担当者あるいは一部のグループなどによる任意的な活動に依存する 同じ部門内でも統一した活動がなされているわけではない それらの内容についても、開発責任者や開発者の知識と経験に基づいて実施されるため、明確な判断基準はない
レベル3	開発を外部発注する場合に、脆弱性対策を目的として、組織的に策定されたセキュリティルールに従った開発を要求する 検収条件として設計書、レビュー結果、セキュリティに関わる機能に特化したテストの実施結果の提出を要求する これらの結果について部門内または部門間で監査する業務フローが規定されている
レベル4	レベル3の取組みに加えて、セキュリティを専門とする部門が存在し、委託先の指導を行う

4.3.5. セキュリティ評価テスト・デバッグ

組込みシステム内の保護すべき資産（データ）を暴露、改ざん、破壊する攻撃は、資産にアクセスする論理的もしくは物理的インタフェースを介して実行される。論理的なインタフェースを介する攻撃としては、例えばバッファオーバフロー攻撃のように、ソフトウェアが想定する範囲外のデータを入力する方法や、システムが使用する制御コードを入力データにいれ込むことにより誤動作を引き起こさせるインジェクション攻撃などがある。

物理的なインタフェースを介する攻撃としては、システムLSIのデバッグ用のピンにデバッガを接続してプログラムを解析するリバースエンジニアリング攻撃や、プログラムの実行とあわせて組込みシステムの状態変化を示す物理量（例えば消費電力）などを観測し、例えば暗号鍵などの機密情報を解析するサイドチャネル攻撃がある。上述したような各種の攻撃手法に対し、脆弱な部分がないか確認する作業がセキュリティ評価テストである。組込みシステムに対する代表的な攻撃手法と、その手法を実行された際の防御動作の検証内容例を以下に示す。

- インジェクション攻撃

  組込みシステム内のDBサーバやシェルなどのプログラムに与えるコマンド文字列を作り出す際に、制御コード等を悪用して、開発者が予期しないコマンドを投入させる手法である。DBサーバやシェルに与えるコマンド列で、制御コード類をエスケープする処理が行われているかを確認することが必要である。管理用のWebインタフェースを備える場合には、IPAが提供するインジェクション検出ツール[16] を用いて確認する。

- バッファオーバフロー

  実行中のプログラムのメモリ（スタックを含む）内に、入力データとして攻撃者が作成した機械語プログラムを送り込んで実行する手法である。バッファオーバフロー対策が行われているか検証するには、入力バッファ長を超えるサイズのデータが入力された場合の処理が正しく記述されているかを確認する。

- DoS攻撃

  組込みシステム内のサーバソフトウェアが所定のサービスを提供できない状態を作り出す攻撃手法である。サーバプログラムが異常終了したり、無限ループに陥ったり、リソースを使い尽くしたりした場合に発生するもので、上記を検出する機構を備えて対策を行う手段が提供されているかを確認する。また、TCP/IPの既知の問題に関しては、IPAが提供する脆弱性検証ツール[17] を用いて確認する。

- リバースエンジニアリング

  組込みシステムにデバッガ等を接続して、動作を解析する攻撃手法である。開発時にデバッグに使用した端子等や、開発者や保守員が使用するデバッグモードは、ユーザが利用できないような対策

---

[16] インジェクション検出ツール（http://www.ipa.go.jp/security/vuln/iLogScanner/index.html）

[17] TCP/IP脆弱性検証ツール（http://www.ipa.go.jp/security/vuln/vuln_TCPIP_Check.html）

を行っているか確認する。

・ サイドチャネル攻撃

組込みシステムの動作を観察することにより、機密情報等を入手・推定する攻撃手法である。重要な機密情報は解析できないように妨害する対策が行われているか確認する。

レベル	取組み概要
レベル1	テストにはセキュリティに関する項目がない
レベル2	担当者の知識・経験に依存した形でセキュリティテスト仕様書が策定される セキュリティ評価の品質は、担当者の知見に委ねられる
レベル3	組織的に策定されたセキュリティテスト仕様書に従って、組織の基準に従った評価を行う 他の機器との接続が行われる場合は、接続対象となる機種との接続性の確認を行う
レベル4	組織的に策定されたセキュリティテスト仕様書に従って、セキュリティ部門が評価を行う 出荷後に対象となる機器に追加・変更が生じた場合は、対象機器との接続性の再確認を行う

4.3.6. ユーザガイド

脆弱性は組込みシステムがユーザの手に渡った後でも発見される。このため、ユーザガイド等を用いて、対応に必要な情報をユーザにあらかじめ通知しておくことが必要である。ユーザガイドに記載すべき内容は以下の通りである。

- ユーザがセキュアに使用するためのガイド

  ネットワークに接続される組込みシステムの場合、内部に記憶されたデータを攻撃者から保護するための手段（例えばパスワードを設定するなど）について説明することが望ましい。また、説明した保護手段をとらなかった場合、どのような事態が想定されるか（例えば、不正アクセスによりファイルが消去される恐れがあるなど）についても説明を加えるとよい。

- トラブル対応手順

  組込みシステムにセキュリティ上の問題が発生した場合、その対応方法（電源オフ、リブートなど）を説明することが望ましい。また、ソフトウェアのアップデートや、発生したトラブルを修復するサービス等を行う場合はその手順等についての説明も必要となる。パスワード等を設定する機能を持つ組込みシステムでは、ユーザがパスワードを忘れた場合の対応や、パスワードを初期化することにより発生する影響について記載することが望ましい。

  また、セキュリティに関連する設定項目について、組込みシステムの出荷時、および初期化した場合に設定されるデフォルト値を記載するとよい。

- 法的な免責事項

  組込みシステムにセキュリティ上の問題が発生し、ユーザに何らかの損害が発生した場合についての免責事項を列挙すべきである。

- 対応しているセキュリティ規格[18)]

  組込みシステムで使用しているセキュリティ規格について、ユーザに通知しておくことが望ましい。セキュリティ規格の例としては、以下のようなものがある。

  - 暗号アルゴリズム（AES、DES、RC4、RSAなど）
  - ハッシュアルゴリズム（SHA-1、SHA-256、MD5など）
  - セキュア通信プロトコル（SSL（TLS）、SSH、IPsecなど）

- 廃棄手順

  ユーザが行うべき機密情報の廃棄手順として、以下を明記することが望ましい。

  - データ消去はユーザの責任で行うべきこと
  - データ消去機能の呼び出し方法
  - データ消去機能の使い方（特に機密情報の特定方法）

---

18) 暗号アルゴリズムに関しては危殆（きたい）化の問題があるため、最新動向に留意して使用することが望ましい。

➢ データ消去の確認方法

レベル	取組み概要
レベル1	ユーザガイドはあるがセキュリティに関する記述はない
レベル2	ユーザガイドにセキュリティに関する記述がある トラブル対応手順と法的な免責事項が明記されている ユーザガイドにセキュリティ要件（危険性と保証範囲）に関する記載がある
レベル3	ユーザガイドにセキュリティルールが反映されているか組織的に検証する トラブル対応手順と法的な免責事項が明記されている ユーザガイドにセキュリティ要件（危険性と保証範囲）に関する記載がある
レベル4	ユーザガイドにセキュリティルールが反映されているか組織的に検証する 新しいセキュリティの脅威に関する更新を定期的に行う トラブル対応手順と法的な免責事項が明記されている

4.3.7. 工場生産管理

組込みシステムは、多くの場合工場で組立てを行う際にソフトウェアの書き込みが行われる。ソフトウェアのみならず、製品1台ごとに割り当てられるシリアル番号シールや、梱包箱に同梱される書類といった、その製品を特定するような情報も該当する。これらは利用形態によってはユーザの個人情報や資産情報に結びつく可能性もある。さらにICカード等の追記可能な記録機能を持つ製品は、半製品として出荷された後に、個人情報を入力する工場も存在することから、生産工程におけるセキュリティ対策も疎かにすることはできない。現場での対策は次の通り。

- 物理的な障壁

  必要とされるセキュリティの程度によって、工場内に物理的な障壁を設けること。その上で人・物が入出する部分にICカード等を用いた入退室管理を行い、あわせてカメラによる監視を行うことが望ましい。

- ネットワークの障壁

  物理的な障壁とあわせて、ネットワークにもファイアウォールやレイヤー2スイッチによるフィルタリング等による障壁を設けることが望ましい。

- ネットワーク・製造機器のログ管理

  入退室管理とともに、ネットワーク・製造機器の稼働・ユーザ・操作履歴を改ざん不可能な形態でログに収集し管理を行うことが望ましい。

- 物品管理

  材料・中間生成物・半完成品・不良品などの個数管理を行うと共に、不正な持ち出しを検知できるようにする。

- 中間生成物の管理・破棄

  製品をチェックする際に用いられるデータ（紙媒体も含む）に機密情報が載せられている場合は、その扱いにも注意することが望ましい。特にトラブル時に関連部署間で機密情報をやり取りせざるを得ない状況を想定し、事前に対応策を策定しておくこと。

<table>
<tr><th>レベル</th><th>取組み概要</th></tr>
<tr><td>レベル1</td><td>特に考慮されない</td></tr>
<tr><td>レベル2</td><td rowspan="3">セキュリティ関連部分については必要に応じて以下の管理が必要となる
・組立作業等を行う機器に対するアクセス制限による、操作者の登録管理（パーティション）
・組立作業等を行うエリア内へのアクセス制限による、操作者の入退室管理（認証）
・入退室管理や持ち込み・持ち出し管理の徹底による、操作者の作業時間管理（時間管理）</td></tr>
<tr><td>レベル3</td></tr>
<tr><td>レベル4</td></tr>
</table>

4.4. 運用フェーズにおける取組み

4.4.1. セキュリティ上の問題への対応

組込みシステムのセキュリティ上の問題点は、主にセキュリティの研究者やユーザが、研究や利用する過程によって発見され、開発元や販売元、あるいは情報処理推進機構に代表される脆弱性の届出機関に報告される。セキュリティ上の問題に対して適切な対処が行われなかった場合、情報の漏えいや故障などユーザにとって不利益が発生し、開発元のブランドイメージが損なわれたり、訴訟に発展したりする可能性も考えられる。そのため、開発元はセキュリティ問題に対して迅速に対応する必要がある。

組込みシステムにセキュリティ上の問題（脆弱性）が発見された場合、開発元はその解決に向けて迅速に行動することが求められる。そのためには、開発元は以下に示す具体例を組織として定める必要がある。

- ユーザあるいは脆弱性関連情報を取扱う機関から情報を受け取るための窓口の設置
- 脆弱性対策のための修正プログラム作成等に向けた対応フローの決定
- 関連諸組織（JPCERT/CC・改修の影響を受ける他ソフトウェアベンダ等）との連絡方法等[19)]

特に窓口の設置に関しては、届けられた脆弱性の情報がいち早く開発者に伝わるように、脆弱性に特化した窓口を作成したり、届けられた情報をハンドリングしたりできる担当者を配置する必要がある。

レベル	取組み概要
レベル1	セキュリティ上の問題（脆弱性）に対応する取組みがなされていない
レベル2	窓口等に届いた脆弱性関連情報は開発部署の担当者に届けられる 脆弱性対策は担当者主導で実施される
レベル3	窓口等に届いた脆弱性関連情報は開発部署の担当者に届けられる 担当者は組織の対応フローに従って、脆弱性対策の実施および脆弱性対策情報の公開の可否の判断を行う
レベル4	脆弱性関連情報を専門的に取扱う窓口および体制（CSIRT：Computer Security Incident Response Team）等や、対応フローが整っており、ユーザや関係諸機関から届けられた脆弱性関連情報はセキュリティ専門組織が管理する 届け出られた脆弱性関連情報に対応するための規定や期日等が策定されており、脆弱性対策情報等に関してはIPAやJPCERT/CCに対して通知する等、公開を行う

[19)] IPAの脆弱性関連情報取扱ガイドライン等を参照のこと。

4.4.2. ユーザへの通知方法と対策方法

組込みシステムに脆弱性が発見され、その修正プログラム等を作成・配布を行う場合、対象となるユーザに対して、修正プログラムやアップデートの適用、場合によっては回収・修理の手段等に関する通知を行う必要がある。特定の企業のみに販売する特殊な組込みシステムであれば、販売先が全て把握できているため、販売先の担当者に通知することで脆弱性対策漏れを防ぐことが可能である。一方で一般向けに販売している組込みシステムにおいては、開発・販売会社が全ての製品のユーザを把握することが難しいため、一人でも多くのユーザに脆弱性対策を促せるための「ユーザへの通知方法」を持つ必要がある。ネットワークに繋げて利用する組込みシステムであれば、定期的に更新情報を検索する仕組みも有効であるが、組込みシステムの利用用途上、ネットワークにつながないことを前提に設計が行われていることもあり、注意が必要である。ほかにも、ユーザに対してWebや葉書等でユーザ登録を行うことを推奨し、登録されたユーザに対して脆弱性対策情報を告知することも有効である。

デジタルテレビのアップデートの場合、深夜や早朝などの利用者の少ない時間帯に、放送波を通じてプログラムのダウンロードが行われる仕組みがある。組込みシステムにこのような仕組みを取り入れることで修正プログラムのスムーズな適応が可能となる。一方で、経済的や環境的な理由でデジタルテレビのコンセントを抜いている場合などにおいては、放送波を通じたプログラムのダウンロードが行えないという問題もある。これらの問題に関してはユーザガイド等で、適切な利用を行うように呼びかける必要がある。

ユーザに通知する際には、脆弱性対策はバグやバージョンアップと違い、ユーザの不利益につながる旨を明確に説明する必要がある。これはユーザがセキュリティ対策と機能追加等のアップデートとの違いを認識し、ユーザ自身が不利益を被ることを回避するためにも、一刻も早い対応を促すためである。

具体的な通知方法としては以下のものが考えられる。

- 郵送
- 電子メール
- 自社のWebページ
- 脆弱性対策情報データベース（JVN）

IPAやJPCERT/CCに脆弱性関連情報を届け出ることにより、脆弱性対策情報データベース（JVN iPedia）や脆弱性対策情報ポータルサイト（JVN）に登録される。脆弱性対策情報データベースでは数多くの脆弱性関連情報を蓄積することを主眼として構築されており、脆弱性対策情報ポータルサイトではタイムリーに脆弱性関連情報を公開するための工夫がなされている。

レベル	取組み概要
レベル1	ユーザには特に通知していない
レベル2	開発・販売組織のWebページなど、ユーザが能動的に脆弱性対策情報を得ることのできる手段を提供している
レベル3	該当する組込みシステムの購入者に対して、メール送付や書面での通知等、購入者へ直接脆弱性対策情報を提供している
レベル4	該当する組込みシステムのユーザが確実に知りえるような手法で、定期的に注意喚起を行う

4.4.3. 脆弱性関連情報の活用

ユーザや関係諸機関によって届けられた脆弱性関連情報は、類似製品への影響や再発防止、次の問題発生時での活用のため、適切に管理する必要がある。度重なる脆弱性の発生は、企業ブランドの低下につながり、システム的な問題だけではなく、経営に直結した問題となり得る。脆弱性関連情報を活用するためには、脆弱性対策プログラムの作り方や配布の仕方だけではなく、「なぜ、その脆弱性が作り込まれてしまったのか」という根本的な問題から検討する必要がある。過去にIPAに届けられた組込みシステムの脆弱性としては、以下のようなものがある。

・クロスサイト・リクエスト・フォージェリの脆弱性
・クロスサイト・スクリプティングの脆弱性
・組込みシステムの持つ証明書に係わる脆弱性
・組込みシステムの初期設定に関する脆弱性

脆弱性関連情報を活用するための手法としては、脆弱性関連情報やその対策を集約したデータベースの作成や、脆弱性関連情報および脆弱性対策を集約・管理するための作業フローの策定、関係者が閲覧するための環境整備等が挙げられる。

具体的には脆弱性対策情報の活用として、以下のような手法が考えられる。

・ 再発防止に向けた上流工程への反映（例：テスト・デバック項目の再構成等）
・ 類似製品に対する脆弱性のチェック

レベル	取組み概要
レベル1	脆弱性関連情報は管理なされていない
レベル2	脆弱性関連情報は担当者等の主導によって管理されている
レベル3	脆弱性対策情報を組織で活用するための業務フローが確立されている
レベル4	脆弱性関連情報およびその対策を組織で活用するための業務フローが確立されており、開発管理と連携したデータベースが構築されている これは組込みシステムの開発関係者も閲覧可能になっている

4.5. 廃棄フェーズにおける取組み

4.5.1. 機器廃棄方法の周知

廃棄の対象となるデータは、購入後ユーザによって蓄積された機密データ（個人情報・プライベートコンテンツ等）である。この機密情報を製品同梱のユーザガイドもしくはWebページの情報によって簡単な操作で消去できることが望ましい。ユーザガイドもしくはWebページにおける記載事項は、下記の４つである。

・ データの消去はユーザの責任において実行すること
・ データ消去機能の呼び出し方法
・ データ消去機能の使い方（特に機密情報の特定方法）
・ データ消去実行後の確認方法

したがって、廃棄時にユーザが機密データを消去できる機能をデフォルトで実装する必要がある。そのためには、ユーザの機密情報の特定方法、廃棄のための技術的手法を実装する必要がある。具体的には、ユーザの機密情報を特定するために、あらかじめ機密情報をその他のデータとは別の領域に格納し、記憶装置には物理的消去機能を実装しなければならない。
例えば、携帯電話のリサイクル過程において、個人情報消去については、下記の取組みを行っている。社団法人電気通信事業者協会「個人情報消去に関する指針」より引用する。

・ 携帯電話等がご不要となった場合に、携帯電話等の内部に保存・蓄積された個人情報をお客様ご自身が確実に消去できるよう端末操作上の仕組みを提供しております。
・ お客様の個人情報の消去に関し、取扱説明書などで分かりやすく説明することに加えて、専売ショップ等において消去操作のお手伝いをさせていただいております。
・ ご不要になった携帯電話等に保存・蓄積された大切なデータをCD-ROMなどにバックアップするサービスも行っております。なお、機種によっては、データの読み出しが不可能な機種もありますので専売ショップ等でご確認ください。
・ ご不要になった携帯電話等をお客様の目の前で破砕処理し、物理的に使用できなくすることも行っております[20]。

レベル	取組み概要
レベル1	廃棄は特に考慮されない
レベル2	廃棄、データ消去などについて、ユーザガイドやwebページなどでユーザに対して情報を公開している
レベル3	組織として廃棄、データ消去、回収について、ユーザガイドやwebページなどでユーザに情報を公開している ユーザに対する公開方法が組織として規定されている
レベル4	組織として廃棄、データ消去、回収について、ユーザガイドやwebページなどでユーザに情報を公開している 製品のトラッキングが行われており、廃棄の際には組織的に対応できる仕組みが考慮されている 組織として積極的に活動を推進するための投資を行っている

[20] 物理的消去機能（破壊）を行う専用工具（ケイタイパンチ）を用意している
（http://www.nttdocomo.co.jp/corporate/csr/activity/hokkaido/natural_env/）

ライフサイクル（マネジメントを含む）		「セキュリティへの取組み」のレベル			
段階(フェーズ)	セキュリティを考慮すべき項目	レベル1	レベル2	レベル3	レベル4
マネジメント	1-1. セキュリティルール (P12)	セキュリティルールの策定は行っていない	セキュリティルールは開発者が自主的に策定している セキュリティルールは開発責任者や担当者主導で策定される	組織としてセキュリティルールを策定し、規則集として明文化している	組織としてセキュリティルールを策定し、規則集として明文化しており、担当者が規則を順守していることを確認するための手段も規定している
	1-2.セキュリティ教育 (P14)	セキュリティ教育は実施していない	セキュリティに関する分科会（小集団活動）が自主的に活動している 必要に応じて開発責任者や開発者主導で知識を習得したり、勉強会が行われたりする	セキュリティ教育が業務に組み込まれている セキュリティに関する講習会や研修の受講が奨励される	セキュリティ教育が業務に組み込まれている 専門家による研修カリキュラムが義務付けられている
	1-3.セキュリティ情報の収集 (P16)	セキュリティ情報は特に収集していない	開発責任者や開発者主導で必要に応じて情報収集を行っている	組織として情報収集を行い、収集結果は開発者が参照可能にしている	脆弱性情報ハンドリングに積極的に関与し、脆弱性関連情報の活用に努めている
企画	2-1.予算 (P17)	セキュリティに係る予算は用意されていない	プロジェクトリーダー（開発責任者）の要求があった場合に、予算確保が容認される 開発責任者や開発者から要求があった場合に、検討される	開発プロセスの一つとしてセキュリティ確保のために一定の基準で予算を割り振ることが前提として組まれている	組織にセキュリティ部門等を設置するための予算を確保し、活動する
開発	3-1.設計 (P18)	設計段階においてセキュリティ対策はほとんど行われず、仕様書類にも記述されない	設計段階で行うべきセキュリティ対策は開発責任者や開発者の判断で実施される 組織としてのルールはなく、実施内容の監査についても明確になっていない	組織として設計段階で行うべきセキュリティ対策のルールが制定されている それに基いた設計プロセスが実施される 実施内容の監査の仕組みはない	組織として設計段階で行うべきセキュリティ対策のルールが制定されている それに基いた設計プロセスが実施され、その内容をセキュリティ部門が監査する仕組みが機能している
	3-2.開発プラットフォーム選定 (P23)	コスト優先、納期優先で選定され、セキュリティについてはほとんど考慮されない	開発時に判明している脆弱性への対策は実施する 出荷後は特に実施されない 開発プラットフォームの選定を実施する担当者または部門がない その製品の開発責任者や開発者によって脆弱性対策の意識が異なり、組織として統一した取組みはなされない	組込みソフトウェアパッケージを利用する場合、販売元からの脆弱性対策プログラム等のリリース情報を監視している 出荷後の製品に対してもソフトウェアアップデート等によって対策を実施している 同じ製品を扱う部門内では脆弱性対策への意識が高く、統一した取組みがなされている	全社的に脆弱性対策を実施している 出荷後の製品に対してもソフトウェアアップデート等によって対策を実施している 販売元の脆弱性対策に関するサポート体制について事前に確認を行う デバイスや回路構成においてセキュリティを考慮した選定を行っている 製品のセキュリティを高める目的で、独自のプラットフォームの開発や調達先への改善要求活動などを実施している
	3-3. ソフトウェア実装 (P25)	セキュリティを意識したコーディングはなされていない ソースコードレビューは実施されない または、実施されてもセキュリティに関する検証はなされていない	脆弱性対策を目的としたコーディングルールやレビューの手法については、担当者あるいは一部グループなどによる任意的な活動に依存する 同じ部門内でも統一した活動がなされているわけではない それらの内容についても、開発責任者や開発者の知識と経験に基づいて実施されるため、明確な判断基準はない	脆弱性対策を目的として、コーディングやレビューを始めとする実装段階におけるルールが組織的に規定されている 担当者は、このルールに則って作業を進める これらの結果について部門内または部門間で監査する業務フローが規定されている	レベル3の取組みに加えて、セキュリティを専門とする部門が存在し、コーディングやレビューにおいて指導を行う また、この組織がルール策定に関与し、不具合とその対策などの情報を基にしてルールの改定等を続けている
	3-4. 開発の外部委託における取組み (P27)	開発を外部委託する場合に、脆弱性対策に関する取組みについては考慮されない セキュリティに関する機能が含まれていても、通常の開発委託と同様の扱いとしている	開発を外部委託する場合の脆弱性対策については、担当者あるいは一部のグループなどによる任意的な活動に依存する 同じ部門内でも統一した活動がなされているわけではない それらの内容についても、開発責任者や開発者の知識と経験に基づいて実施されるため、明確な判断基準はない	開発を外部委託する場合に、脆弱性対策を目的として、組織的に策定されたセキュリティルールに従った開発を要求する 検収条件として設計書、レビュー結果、セキュリティに関わる機能に特化したテストの実施結果の提出を要求する これらの結果について部門内または部門間で監査する業務フローが規定されている	レベル3の取組みに加えて、セキュリティを専門とする部門が存在し、委託先の指導を行う
	3-5.セキュリティ評価テスト・デバッグ (P28)	テストにはセキュリティに関する項目がない	担当者の知識・経験に依存した形でセキュリティテスト仕様書が策定される セキュリティ評価の品質は、担当者の知見に委ねられる	組織的に策定されたセキュリティテスト仕様書に従って、組織の基準に従った評価を行う 他の機器との接続が行われる場合は、接続対象となる機種との接続性の確認を行う	組織的に策定されたセキュリティテスト仕様書に従って、セキュリティ部門が評価を行う 出荷後に対象となる機器に追加・変更が生じた場合は、対象機器との接続性の再確認を行う
	3-6.ユーザガイド (P30)	ユーザガイドはあるがセキュリティに関する記述がない	ユーザガイドにセキュリティに関する記述がある トラブル対応手順と法的な免責事項が明記されている ユーザガイドにセキュリティ要件（危険性と保証範囲）に関する記載がある	ユーザガイドにセキュリティルールが反映されているか組織的に検証する トラブル対応手順と法的な免責事項が明記されている ユーザガイドにセキュリティ要件（危険性と保証範囲）に関する記載がある	ユーザガイドにセキュリティルールが反映されているか組織的に検証する 新しいセキュリティの脅威に関する更新を定期的に行う トラブル対応手順と法的な免責事項が明記されている
	3-7.工場生産管理 (P32)	特に考慮されない	セキュリティ関連部分については必要に応じて以下の管理が必要となる ・組立作業等を行う機器に対するアクセス制限による、操作者の登録管理（パーティション） ・組立作業等を行うエリア内へのアクセス制限による、操作者の入退出管理（認証） ・入退出管理や持ち込み・持ち出し管理の徹底による、操作者の作業時間管理（時間管理）		
運用	4-1. セキュリティ上の問題への対応 (P33)	セキュリティ上の問題（脆弱性）に対応する取組みがなされていない	窓口等に届いた脆弱性関連情報は開発部署の担当者に届けられる 脆弱性対策は担当者主導で実施される	窓口等に届いた脆弱性関連情報は開発部署の担当者に届けられる 担当者は組織の対応フローに従って、脆弱性対策の実施および脆弱性対策情報の公開の可否の判断を行う	脆弱性関連情報を専門的に取扱う窓口および体制（CSIRT ：Computer Security Incident Response Team）等や、対応フローが整っており、ユーザや関係緒機関から届けられた脆弱性関連情報はセキュリティ専門組織が管理する 届け出られた脆弱性関連情報に対応するための規定や期日等が策定されており、脆弱性対策情報等に関してはIPAやJPCERT/CCに対して通知する等、公開を行う
	4-2.ユーザへの通知方法と対策方法 (P34)	ユーザには特に通知していない	開発、販売組織のWebページなど、ユーザが能動的に脆弱性対策情報を得ることのできる手段を提供している	該当する組込みシステムの購入者に対して、メール送付や書面での通知等、購入者へ直接脆弱性対策情報を提供している	該当する組込みシステムのユーザが確実に知りえるような手法で定期的に注意喚起を行う
	4-3.脆弱性情報の活用 (P35)	脆弱性関連情報は管理されていない	脆弱性関連情報は担当者等の主導によって管理されている	脆弱性関連情報を組織で活用するための業務フローが確立されている	脆弱性関連情報およびその対策を組織で活用するための業務フローが確立されており、開発管理と連携したデータベースが構築されている これは組込みシステムの開発関係者も閲覧可能になっている
廃棄	5-1.機器廃棄方法の周知 (P36)	廃棄は特に考慮されない	廃棄、データ消去などについて、ユーザガイドやwebページなどでユーザに対して情報を公開している	組織として廃棄、データ消去、回収などについて、ユーザガイドやwebページなどでユーザに情報を公開している ユーザに対する公開方法が組織として規定されている	組織として廃棄、データ消去、回収などについて、ユーザガイドやwebページなどでユーザに情報を公開している 製品のトラッキングが行われており、廃棄の際には組織的に対応できる仕組みが考慮されている 組織として積極的に活動を推進するための投資を行っている